



**Numerical EDA**

In this chapter, you'll be working with a dataset obtained from the [**UCI Machine Learning Repository**](https://archive.ics.uci.edu/ml/datasets/Congressional+Voting+Records) consisting of votes made by US House of Representatives Congressmen. Your goal will be to predict their party affiliation ('Democrat' or 'Republican') based on how they voted on certain key issues. Here, it's worth noting that we have preprocessed this dataset to deal with missing values. This is so that your focus can be directed towards understanding how to train and evaluate supervised learning models. Once you have mastered these fundamentals, you will be introduced to preprocessing techniques in Chapter 4 and have the chance to apply them there yourself - including on this very same dataset!

Before thinking about what supervised learning models you can apply to this, however, you need to perform Exploratory data analysis (EDA) in order to understand the structure of the data. For a refresher on the importance of EDA, check out the first two chapters of [**Statistical Thinking in Python (Part 1)**](https://www.datacamp.com/courses/statistical-thinking-in-python-part-1).

Get started with your EDA now by exploring this voting records dataset numerically. It has been pre-loaded for you into a DataFrame called df. Use pandas' .head(), .info(), and .describe() methods in the IPython Shell to explore the DataFrame, and select the statement below that is **not** true.

**Instructions**

**50 XP**

The DataFrame has a total of 435 rows and 17 columns.

Except for 'party', all of the columns are of type int64.

The first two rows of the DataFrame consist of votes made by Republicans and the next three rows consist of votes made by Democrats.

There are 17 *predictor variables*, or *features*, in this DataFrame.

The target variable in this DataFrame is 'party'.

 +50 XP

Great work! The number of columns in the DataFrame is not equal to the number of features. One of the columns - 'party' is the target variable.